Preprint
UCRL-JC-144075

A Load Balanced Domain Decomposition Method using Wavelet Analysis

L. Jameson, J. Johnson, and J. Hesthaven

This article was submitted to the
Joint Symposium on Parallel Programming, June 5 – 8, 2001,
Kyoto, Japan

May 31, 2001

A Load Balanced Domain Decomposition Method Using Wavelet Analysis

Leland Jameson, John Johnson

Lawrence Livermore National Laboratory
University of California
P.O. Box 808
Livermore, CA 94551

Jan Hesthaven

Division of Applied Mathematics
Brown University
Providence, RI

Abstract

Wavelet Analysis provides an orthogonal basis set which is localized in both the physical space and the Fourier transform space. We present here a domain decomposition method that uses wavelet analysis to maintain roughly uniform error throughout the computation domain while keeping the computational work balanced in a parallel computing environment.

Contents

List of Figures

1 Introduction

The ability of wavelets to accurately and efficiently represent functions with localized features [1, 2, 3] has spawned intensive research into applying wavelets to the solution of partial differential equations. Traditionally, this effort has been centered around using wavelets as an orthogonal and complete basis, spanning a space of approximate solutions satisfying an equation in a Galerkin sense [4, 5, 6]. Besides the well known difficulties associated with such an approach for non-linear problems, there are also challenges to handling non-trivial boundary conditions in an accurate and stable manner.

Such restrictions on the applicability of wavelet based methods for the solution of problems of more general interest have, in recent years, induced significant interest into grid-based collocation wavelet methods, [4, 6, 7, 8, 9]. The formulation and implementation of multi-dimensional pure wavelet collocation methods, however, remains a challenging task.

In the present work we take a somewhat different approach to arrive at a grid based method exploiting the unique properties of wavelets. Rather than using the wavelets as a basis, we utilize the ability of wavelets to not only detect the existence of high-frequency information but also to supply information about the spatial location of strongly inhomogeneous regions. Such a region would, in the Galerkin formulation, require one to use many wavelet coefficients while, within a collocation formulation, it would require a very fine grid.

The gap between wavelets and finite difference schemes may seem rather large. However, a very close connection between these two techniques has recently been established [10, 11, 12]. It has been advocated that wavelets be used for grid generation and order selection only, while the scheme for solving the partial differential equation be based solely on finite difference schemes defined on variable grids. In particular, the differentiation operators associated with wavelet based collocation methods are in some cases equivalent to operators appearing from variable grid finite difference operators [13, 14]. This suggests that wavelet analysis provides the information required to construct adaptive finite difference schemes on arbitrary grids with the error estimation based on the wavelet analysis. This interpretation alleviates problems associated with non-linear terms and,to some extent, by using one-sided finite difference schemes, problems with arbitrary boundary conditions as well. We shall return to the connection between collocation wavelet methods and finite difference schemes in the following section.

Let us now, however, simply claim that wavelets provide the proper tool

for the formulation of adaptive, arbitrary grid finite difference schemes and consider the difficulties associated with taking such an approach. Besides the natural difficulties of implementing an arbitrary grid and arbitrary order multi-dimensional finite difference method, finite difference schemes defined on arbitrary grids are known to introduce numerical artifacts [7, 6], resulting in an amplification of numerical noise. As a consequence, coarsening in smooth regions of the solution is a less than trivial task – particularly when considering the use of high-order methods. Moreover, wavelets are best suited for application on equidistant grids which, for problems beyond one dimension, suggests a tensor-product approximation. This, on the other hand, makes the application of such methods hard for problems in complex domains.

The requirement for a semi-structured grid, and the need for geometric flexibility, points toward the introduction of a multi-domain formulation as an appropriate way of progressing. Indeed, as has been realized over the last decade within the spectral methods community [15, 16, 17], multi-domain methods alleviate many of the problems associated with the use of high order methods in complex geometries. In this work, we propose to combine the geometric flexibility and computational efficiency of a multi-domain scheme with the adaptivity, facilitated by the wavelet analysis and the associated finite difference operators, to arrive at a scheme which circumvents most of the problems discussed above while providing a natural load-balanced data-decomposition within a parallel framework.

The remaining part of this paper is organized as follows. In Section 2 we discuss the relation between finite-difference methods on arbitrary grids and the wavelet decomposition based on Daubechies wavelets. To extend the wavelet based grid and order adaptation to multi-dimensional and geometrically complex problems, we find it necessary to introduce a minimum amount of structure into the global grid. These issues are addressed in Section 3 where we propose to combine the wavelet analysis with a multi-domain formulation such as to alleviate various problems hitherto associated with wavelet based methods. A performance analysis is given in Section 4 and Section 5 contains a few concluding remarks.

2 Wavelet Analysis

Possibly the most instructive way to think of wavelets is in contrast to traditional analysis techniques such as Fourier analysis. With Fourier analysis we

analyze discrete or continuous data using basis functions which are global, smooth and periodic. This analysis yields a set of coefficients, say, a_k, which gives the amount of energy in the data at frequency k. Wavelet analysis, by contrast, analyzes data with basis functions which are local, slightly smooth, not periodic, and which vary with respect to scale and location. Wavelet analysis thereby produces a set of coefficients $b_{j,k}$ which give the amount of energy in the data at scale j and location k. Wavelet analysis can serve as a good complement to Fourier analysis. In fact, data which is efficiently analyzed with Fourier analysis often is not efficiently analyzed with wavelet analysis and the opposite situation also holds.

For our purposes here we will confine our discussion to the so-called orthogonal wavelets and specifically the Daubechies family of wavelets. The orthogonality property leads to a clear indication when data deviates from a low-order polynomial, the importance of which will become clear when we discuss numerical methods.

2.1 Theoretical Background in the Continuous World

To define Daubechies-based wavelets, [Daubechies (1988) and Erlebacher (1996)], consider the two functions $\phi(x)$, the scaling function, and $\psi(x)$, the wavelet. The scaling function is the solution of the dilation equation,

$$\phi(x) = \sqrt{2} \sum_{k=0}^{L-1} h_k \phi(2x - k), \tag{1}$$

which carries the name "dilation equation" since the independent variable x appears alone on the left hand side but is multiplied by 2, or dilated, on the right hand side. One also requires the scaling function $\phi(x)$ be normalized: $\int_{-\infty}^{\infty} \phi(x)dx = 1$. The wavelet $\psi(x)$ is defined in terms of the scaling function,

$$\psi(x) = \sqrt{2} \sum_{k=0}^{L-1} g_k \phi(2x - k). \tag{2}$$

One builds an orthonormal basis from $\phi(x)$ and $\psi(x)$ by dilating and translating to get the following functions:

$$\phi_k^j(x) = 2^{-\frac{j}{2}} \phi(2^{-j}x - k), \tag{3}$$

and

$$\psi_k^j(x) = 2^{-\frac{j}{2}} \psi(2^{-j}x - k), \tag{4}$$

where $j, k \in Z$. j is the dilation parameter and k is the translation parameter.

It is usual to let the spaces spanned by $\phi_k^j(x)$ and $\psi_k^j(x)$ over the parameter k, with j fixed, be denoted by V_j and W_j respectively,

$$V_j = \underset{k \in Z}{span}\ \phi_k^j(x), \tag{5}$$

$$W_j = \underset{k \in Z}{span}\ \psi_k^j(x). \tag{6}$$

The spaces V_j and W_j are related by,

$$... \subset V_1 \subset V_0 \subset V_{-1} \subset ..., \tag{7}$$

and

$$V_j = V_{j+1} \bigoplus W_{j+1}, \tag{8}$$

where the notation $V_0 = V_1 \bigoplus W_1$ indicates that the vectors in V_1 are orthogonal to the vectors in W_1 and the space V_0 is simply decomposed into these two component subspaces.

The coefficients $H = \{h_k\}_{k=0}^{L-1}$ and $G = \{g_k\}_{k=0}^{L-1}$ are related by $g_k = (-1)^k h_{L-k}$ for $k = 0, ..., L-1$. All wavelet properties are specified through the parameters H and G. If one's data is defined on a continuous domain such as $f(x)$ where $x \in R$ is a real number then one uses $\phi_k^j(x)$ and $\psi_k^j(x)$ to perform the wavelet analysis. If, on the other hand, one's data is defined on a discrete domain such as $f(i)$ where $i \in Z$ is an integer then the data is analyzed, or filtered, with the coefficients H and G. In either case, the scaling function $\phi(x)$ and its defining coefficients H detect localized low frequency information, i.e., they are low-pass filters (LPF), and the wavelet $\psi(x)$ and its defining coefficients G detect localized high frequency information, i.e., they are high-pass filters (HPF). Specifically, H and G are chosen so that dilations and translations of the wavelet, $\psi_k^j(x)$, form an orthonormal basis of $L^2(R)$ and so that $\psi(x)$ has M vanishing moments which determines the accuracy. In other words, $\psi_k^j(x)$ will satisfy

$$\delta_{kl}\delta_{jm} = \int_{-\infty}^{\infty} \psi_k^j(x)\psi_l^m(x)dx, \tag{9}$$

where δ_{kl} is the Kronecker delta function, and the accuracy is specified by requiring that $\psi(x) = \psi_0^0(x)$ satisfy

$$\int_{-\infty}^{\infty} \psi(x)x^m dx = 0, \tag{10}$$

for $m = 0, ..., M-1$. This statement on accuracy can be explained in an alternative manner. Recall that within the scaling function subspaces V_j one can reconstruct low order polynomials exactly up to a given order so that x^0, x^1,..., x^{M-1} can be represented exactly by appropriately choosing scaling function coefficients. Accordingly, the subspace W_j which is orthogonal to V_j and consequently the basis elements $\psi_j(x)$ of W_j will be orthogonal to all elements contained in V_j such as these low-order polynomials. So, one can see that by specifying the number of vanishing moments of the wavelets one has in effect specified the number of polynomials which can be represented exactly and hence the numerical accuracy of the method.

For representing functions in $L^2(R)$ one can see from the above expressions that for any function $f(x) \in L^2(R)$ there exists a set $\{d_{jk}\}$ such that

$$f(x) = \sum_{j \in Z} \sum_{k \in Z} d_k^j \psi_k^j(x), \tag{11}$$

where

$$d_k^j = \int_{-\infty}^{\infty} f(x) \psi_k^j(x) dx. \tag{12}$$

The two sets of coefficients H and G are known as quadrature mirror filters. For Daubechies wavelets the number of coefficients in H and G, or the length of the filters H and G, denoted by L, is related to the number of vanishing moments M by $2M = L$. For example, the famous Haar wavelet is found by defining H as $h_0 = h_1 = 1$. For this filter, H, the solution to the dilation equation (1), $\phi(x)$, is the box function: $\phi(x) = 1$ for $x \in [0, 1]$ and $\phi(x) = 0$ otherwise. The Haar function is very useful as a learning tool, but because of its low order of approximation accuracy and lack of differentiability it is of limited use as a basis set. The coefficients H needed to define compactly supported wavelets with a higher degree of regularity can be found in Daubechies (1988). As is expected, the regularity increases with the support of the wavelet. The usual notation to denote a Daubechies-based wavelet defined by coefficients H of length L is D_L.

2.2 Practical Implementation in the Discrete World

Naturally, infinite sums and integrals are meaningless when one begins to implement a wavelet expansion on a computer. One must find appropriate ways to implement discrete counterparts to the continuous operations which were outlined in the previous subsection. That is, nothing is continuous on a computer and since wavelet analysis was derived in the continuous

world of differential and integral mathematics is it necessary to consider a discrete version of the above continuous theory. Generally, operations such as integration are easily approximated with an appropriate order quadrature formula, but one would like to use as few quadratures as possible to limit the number of approximations which are made. We will see in this section how we can easily perform all the wavelet decompositions with relatively few approximations.

In a continuous wavelet expansion, functions with arbitrarily small-scale structures can be represented. In practice, however,depending on the numerical grid resolution or the sampling frequency in a signal processing scenario, there is a limit to how small the smallest structure can be. Hence, on a computer an approximation would be constructed in a finite space such as

$$V_0 = W_1 \oplus W_2 \oplus \ldots \oplus W_J \oplus V_J,$$

with the approximation being

$$P_{V_0} f(x) = \sum_{k \in Z} s_k^J \phi_k^J(x) + \sum_{j=1}^{J} \sum_{k \in Z} d_k^j \psi_k^j(x) \quad , \tag{13}$$

with

$$d_k^j = \int_{-\infty}^{\infty} f(x) \psi_k^j(x) dx,$$

$$s_k^J = \int_{-\infty}^{\infty} f(x) \phi_k^J(x) dx$$

utilizing orthogonality. Within this expansion, the scale $j = 0$ is arbitrarily chosen as the finest scale required, and scale J would be the scale at which a kind of local average, $\phi_k^J(x)$, provides sufficient large scale information, i.e. the first term in Eq.(13) provides the local mean around which the function oscillates.

One must also limit the range of the location parameter, k Assuming periodicity of $f(x)$ implies periodicity on all wavelet coefficients, s_k^j and d_k^j, with respect to k. For the non-periodic case, since k is directly related to the location, a limit is imposed on the values of k when the location being addressed extends beyond the boundaries of the domain.

The wavelet decomposition matrix is the matrix embodiment of the dilation equation, Eq.(1), defining the scaling function and the accompanying equation defining the wavelet, Eq.(2). The following two recurrence relations

for the coefficients, s_k^j and d_k^j, in Eq.(13) are given as

$$s_k^j = \sum_{n=1}^{L} h_n s_{n+2k-2}^{j-1} \ , \ d_k^j = \sum_{n=1}^{L} g_n s_{n+2k-2}^{j-1} \ ,$$

as obtained from Eqs.(1)-(2), and we recall that h_n refers to the chosen filter while we have $g_n = -(-1)^n h_{L-n}$.

Denote the decomposition matrix embodied by these two equations, assuming periodicity, by $P_N^{j,j+1}$ where the matrix subscript denotes the size of the square matrix while the superscripts indicate that P is decomposing from scaling function coefficients at scale j to scaling function and wavelet function coefficients at scale $j+1$, i.e. $P_N^{j,j+1}$ maps $\vec{s}_j$ onto $\vec{s}_{j+1}$ and $\vec{d}_{j+1}$:

$$P_N^{j,j+1} : \left[\ \vec{s}_j \ \right] \rightarrow \left[\begin{array}{c} \vec{s}_{j+1} \\ \vec{d}_{j+1} \end{array} \right] \ , \tag{14}$$

where we by $\vec{s}_j$ refer to the vector containing the coefficients at scale j. Note that the vectors at scale $j+1$ are half as long as the vectors as scale j.

Suppose, for illustration, the wavelet being used is the four coefficient D_4 wavelet, and that one wants to project from 8 scaling function coefficients at scale j to 4 scaling function coefficients at scale $j+1$ and 4 wavelet coefficients at scale $j+1$. The decomposition matrix for the case of periodic boundary conditions, $P_8^{j,j+1}$, thus becomes

$$P_8^{j,j+1} \equiv \left[\begin{array}{cccccccc} h_1 & h_2 & h_3 & h_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_1 & h_2 & h_3 & h_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_1 & h_2 & h_3 & h_4 \\ h_3 & h_4 & 0 & 0 & 0 & 0 & h_1 & h_2 \\ g_1 & g_2 & g_3 & g_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & g_1 & g_2 & g_3 & g_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & g_1 & g_2 & g_3 & g_4 \\ g_3 & g_4 & 0 & 0 & 0 & 0 & g_1 & g_2 \end{array} \right] , \tag{15}$$

where the periodicity is reflected in the coefficients being wrapped around. If the boundary conditions are not periodic, then one must extend the data and the scaling function coefficients at every scale in a "smooth" manner. This involves extending the scaling function data with a polynomial one order higher than the number of vanishing moments of the wavelet. But, for the purpose of understanding the method, it is sufficient to study the periodic matrix such as the one that we have created above. It is a matrix

such as $P^{j,j+1}$ that is applied directly to the data and then directly to each level of scaling function coefficients. To be perfectly correct one would first approximate the scaling function coefficients at the finest scale using the raw data, however, in practice it seems to make very little difference if one simply considers the raw data to be the scaling function coefficients. So, for our purposes here we will simply use the raw data as the scaling function coefficients on the finest scale. The repeated application of the matrix $P^{j,j+1}$ yields the wavelet coefficients at the various scales, and it is these wavelet coefficients that provide a guide to the errors committed during the numerical calculation, as will be further illustrated in the next paragraph and explained precisely in the next section.

Let us consider that the raw data is given and it is assumed to be the scaling function coefficients on the finest scale, $\vec{s}_0$. One wavelet decomposition yields the scaling function coefficients and wavelet coefficients at scale $j = 1$, $\vec{s}_1$ and $\vec{d}_1$. A second application of the wavelet decomposition matrix will yield the vectors $\vec{s}_2$ and $\vec{d}_2$. It is the vectors $\vec{d}_1$, $\vec{d}_2$, ... which yield the critical information on the numerical errors. If, for example, one sees that the values of the $\vec{d}_1$ are relatively large in the middle of the vector, then it is clear that within this one dimensional vector the largest errors will be in the middle of the one dimensional domain from which this vector was derived.

3 A Wavelet Based Multi-Domain Scheme

In the first part of this paper, we discussed several advantages to our approach of exploiting the close connection between finite-difference methods and wavelets so that wavelets are used to adaptively select grids while a finite-difference methodology is used to compute derivatives. By restricting the computation between the wavelet-based adaptation phases so that the product of the number of grid points in a subdomain and the order used in the subdomain remains constant, our method maintains an equal computational load on all subdomains and is thus optimally load balanced. In addition, since the computation is performed in the physical space rather than the transform space, non-linear terms and finite computational domains pose no significant problems.

So far, the discussion has focused on equidistant or Chebyshev grids. The multidimensional extension of this approach uses tensor products which require the computational domain to be diffeomorphic to the unit square /cube thereby limiting the type of problems to which this approach can be applied.

There are several ways to circumvent this restriction. For example, one may simply embed the general computational domain into a simple rectangular domain approximating the boundaries using stair-casing. While this approach works well with low-order finite difference methods, it causes severe problems with high-order methods. This has prompted us to combine the high spatial accuracy and adaptivity of simple domains with the geometric flexibility provided by a multi-domain formulation.

In this scenario, the geometrically complex computational domain is decomposed into a number of simple geometric building blocks, (e.g. quadrilaterals/hexahedrals), in which a tensor product formulation can be straightforwardly applied. This offers geometric flexibility and lends itself to a parallel implementation. A discussion of additional advantages, problems, and general methodology of multi-domain schemes can be found in [15, 16, 17].

Once we have completed the domain decomposition, we can apply the ideas of the previous section to each subdomain. That is, we can perform the wavelet analysis and adaptivity within each subdomain.

As an illustration, let us consider the solution of the linear two-dimensional wave equation

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = 0 \ , \ \ (x, y) \in [0, 2]^2 \ , \tag{16}$$

where $u = u(x, y, t)$ and the initial conditions are taken to be a Gaussian pulse of the form

$$u(x, y, 0) = \exp\left[-\frac{(x - x_0)^2}{\delta_x^2} - \frac{(y - y_0)^2}{\delta_y^2} \right] \ ,$$

where (x_0, y_0) signifies the center of the pulse and (δ_x, δ_y) the variance along x and y, respectively. The exact solution is given by convecting the initial condition with the velocity $(1, 1)$.

Now, let the 2D decomposition result in 16 equally sized squares, see Fig. (1). In each domain, we solve the equation on an adaptive grid maintaining a 4th order scheme. Grid adaptation can be performed using full adaptation within each block where the grids within each block may be completely unstructured or block-wise adaptation where the grids within each block remain regular but may vary between adjacent subdomains. While the first option is more general, it is nontrivial to implement such a scheme efficiently. Refining only on a block-wise course-grain skeleton, retains the structure within each subdomain while allowing a significant degree of adaptation.

After several timesteps, wavelets are used to detect regions of the computational domain which contain small structure. The scale of this information is determined by the magnitude of the wavelet coefficients. Once these regions are discovered, the number of grid points and order are adjusted accordingly, while keeping the product of the grid points and order constant (fig. 1). In other words wavelets are used to keep the L_∞ error roughly uniform throughout the domain. Instead of adjusting the grid density and numerical order at each point, it is done block by block.

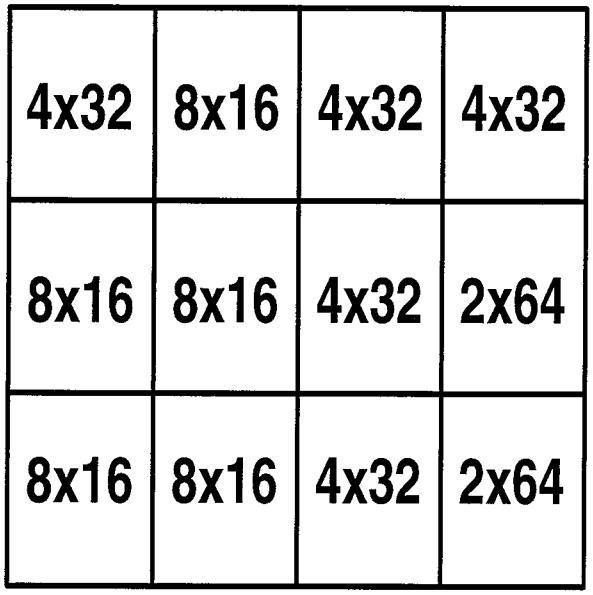

Figure 1: Subdomains with equal computational granularity.

Patching of the scalar wave equation, Eq.(16), is performed by communicating information across boundaries along the direction of propagation, i.e. out-flowing information from one subdomain enters the adjacent subdomain as inflow/boundary conditions.

One of the challenges of multi-domain methods is to ensure that the correct global solution is achieved by solving a number of smaller local problems. The construction of proper patching remains an area of active research. Solution are typically problem specific. The adaptive framework that we have set up here, however, is independent of the specifics of patching schemes and the problem being solved, since the wavelet analysis essentially is applied as in a signal analysis approach.

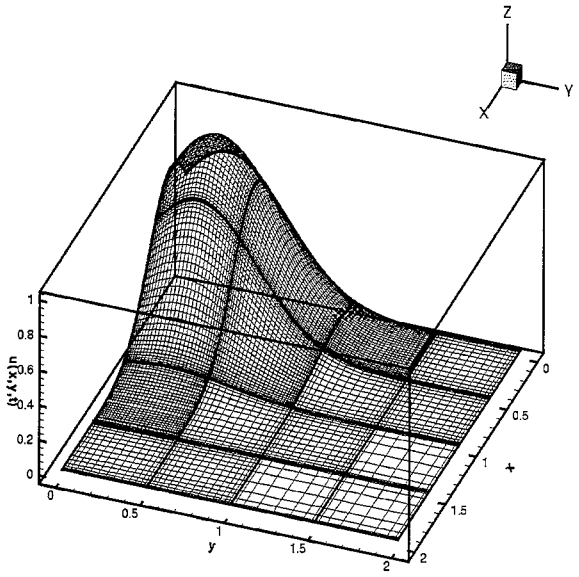

Figure 2: Gaussian Pulse Propagating across domain.

Figure (2) shows the Gaussian pulse propagating diagonally across the domain. One can see the changing grid point density. As the grid becomes more dense, the computational order is decreased thereby keeping the work in each subdomain fixed.

The observant reader may notice that the subdomain grids in Fig. (1) are non-uniform. Indeed, when using a 4th order scheme in each subdomain there is no reason a uniform grid could not be used provided it is terminated in a stable manner using a 3rd order one sided stencil. However, as discussed in [11] there is no reason why one cannot also adapt the order of the scheme used in each subdomain, employing high order schemes in regions with course grids, reflecting smooth solutions, and low order schemes in regions with great variation and very fine grids. In order to do so, i.e. to use schemes of order higher than 4, we must however cluster the grids to maintain stability. The error estimator is found to yield reliable estimates provided L_∞ is set to scale with the order of the scheme, reflecting the higher regularity assumed to exist when using high order schemes.

Adapting order as well as grid density has, aside from the numerical advantages of high-order methods in connection with long-time integration, the potential of offering optimal load-balancing in a parallel setting as the num-

ber of grid points times the order of the scheme, providing an approximate measure for the overall work, can be kept close to constant. Furthermore, one of the goals of any numerical method is to minimize the error for a given computational cost. Fundamentally, this means approximating the data effectively with low-order local polynomial approximations, the error being given by the truncation error. In other words, rough localized features are approximated most effectively by low order polynomials with a high density of grid points while smooth large-scale features are approximated most effectively by high order polynomials on a coarse grid. For this reason, we increase the order of the numerical method as we decrease the grid point density. In addition to maximizing the computational efficiency, the work remains constant in all the subdomains.

4 Performance Analysis

To analyze the predicted performance of this technique, we note the total computation time, T_{tot}, can be described as,

$$T_{tot} = T_{comp} + T_{comm} \tag{17}$$

where T_{comp} is the total time spent in the computation phase and T_{comm} is the total time spent in processor communication. T_{comm} is composed of two components, T_s and T_w where T_s is the startup time to initiate interprocessor communication and T_w is the time to transmit a single word, see [19]. Thus the total communication time is:

$$T_{comm} = T_s + L * T_w \tag{18}$$

where L is the number of words to be transmitted. The computation time is composed of two components T_k, T_{wt} where T_k is the kernel computation or the actual time spent calculating the derivative on the grid points, and T_{wt} is the time to compute the wavelet transform.

The number of floating point operations spent by a single node on the derivative computation is $G_i * O_i$ where G_i is the number of grid points at node i and O_i is the order of computation at node i. (This is not entirely accurate. Depending upon the order there may be additional terms. For the sake of brevity we will ignore these terms since they don't substantively affect the resulting analysis of the method.) What this technique proposes is to keep $G_i * O_i = K$ over all the nodes where K is a constant which we are calling the *computational granularity.*

Now at each iteration only one floating point value needs to be communicated across the boundary to an adjacent node. Furthermore, the wavelet transform is computed locally thus L in the communication equation is equal to exactly 1 so

$$T_{comm} = T_s + T_w. \tag{19}$$

For now, lets ignore T_{wt} and consider the relationship between T_k and T_{comm}. The method will be computation bound iff $T_k >> T_{comm}$. Similarly it will be communication bound iff $T_{comm} >> T_k$. But $T_k = K/FlopRate$. For $T_k >> T_{comm}$, $K >> FlopRate * (T_s + T_w)$. Using typical values for an IBM SP2, see [19], an example bound for K can be approximated as $K >> 200Mflop/s * (4.01ms + 0.11ms)$ or $K >> 840$. Recall, since $K = G_i * O_i$ and since O is typically in the range $[0, 4]$, for values of $G_i >> 210$, this method is computation bound. This is not an unreasonable value for scientific computations so in most cases the method won't be communication bound.

To compute the effect of T_{wt}, we note that the cost of the wavelet transform is O(G) where G is the total number of grid points (or $O(\sum_{i=1}^{maxNodes} G_i)$.

The wavelet transform does not need to be calculated every timestep and as the number of timesteps between transforms increases, the cost of the wavelet transform will becomes less significant. Since in the transform phase each node is computing the full wavelet transform and since the computation phase is load-balanced by keeping the computational granularity constant, the complete method will be perfectly load balanced.

5 Conclusion

The purpose of this work has been two fold. In the first part we set out to show the close connection between differentiation based on the use of Daubechies wavelets and that of traditional centered finite difference schemes. Indeed, the capability for data compression, being the main argument for the use of wavelet methods, manifests itself in finite difference methods as the possibility for the use of variable grid schemes. Hence, we concluded, based on the above connection as well as a careful discussion of the problems associated with pure wavelet methods, that the proper way of using the wavelets is for identifying exactly where to refine and coarsen the computational grids to maintain a given accuracy, while the well known finite difference framework should be chosen for actually computing derivatives. Besides from the intuitive ease of the grid based approach, it also

offers advantages when the need to deal with boundary conditions or non-linear terms arises.

Extending the wavelet optimized finite difference methods to multi dimensional problems involves the introduction of tensor product grids with the resulting loss of geometric flexibility. To overcome this, we showed how to use a multi-domain formulation in which each geometrically simple subdomain is being dealt with in a straightforward extension of the one-dimensional framework, while the multi-domain setting provides a global skeleton that makes the implementation less troublesome. The block adaptivity proposed here yields significant savings even for a problem as simple as the linear wave equation while eliminating several problems hitherto associated with wavelet optimized finite difference schemes. Moreover, an order and grid adaptive scheme provides advantages in terms of load balancing with a parallel setting.

The generalization of the present framework to problems of more complicated character, i.e. problems of electromagnetics and acoustics, poses no significant algorithmic problems and we hope to report on such development in the near future.

Since one of the goals of this method is to achieve near uniform error, an extension of this study would be to compute the exact error introduced when considering variations in the time step, the number of time steps between wavelet analyses and granularity of the domain decomposition.

We are currently implementing this scheme in a massively parallel environment to verify predicted performance with quantitative results.

References

[1] Y. Meyer, *Ondelettes et Operators.* Hermann, Paris, 1990.

[2] I. Daubechies, *Ten Lectures on Wavelets.* CBMS-NSF **61**, Society of Industrial and Applied Mathematics, Philadelphia, 1992

[3] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Cambridge, 1996.

[4] A. Harten, *Multiresolution Representation and Numerical Algorithms: A Brief Review.* ICASE Report No. 94-59, NASA Langley Research Center, Hampton. 1994.

[5] G. Erlebacher, M.Y. Hussaini and L. Jameson (Eds), *Wavelets: Theory and Applications.* Oxford University Press, Oxford, 1996.

[6] J. Waldén, *Wavelet Solvers for Hyperbolic PDE's*, Acta Univ. Ups. *Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology* **240**. Uppsala, Sweden, 1996

[7] L. Jameson, *Wavelets and Numerical Methods.* PhD Thesis, Division of Applied Mathematics, Brown University, 1993 (unpublished).

[8] M. Holmström, *Solving Hyperbolic PDE's Using Interpolating Wavelets.* Report No. 189/1996, Department of Scientific Computing, Uppsala University, Sweden. 1996.

[9] W. Cai and J. Z. Wang, *Adaptive Wavelet Collocation Methods for Initial Value Boundary Problems of Nonlinear PDE's*, SIAM J. Numer. Anal. **33**(1996), pp. 937-970.

[10] L. Jameson, *On the Wavelet-Optimized Finite Difference Method*, ICASE Report No. 94-9, NASA Langley Research Center, Hampton. 1994.

[11] L. Jameson, *A Wavelet-Optimized, Very High Order Adaptive Grid and Order Numerical Method*, SIAM J. Sci. Comput. 1997 - to appear.

[12] L. Jameson, *Wavelet-based Grid Generation*, Appl. Numer. Anal. 1997 - to appear.

[13] L. Jameson, *On The Wavelet Based Differentiation Matrix*, J. Sci. Comput. **8**(1993), pp. 267-305.

[14] L. Jameson, *On The Differentiation Matrix for Daubechies-Based Wavelets on an Interval*, SIAM J. Sci. Comput. **17**(1996), pp. 498-516.

[15] C. Canuto, M.Y. Hussaini, A. Quarteroni and T.A. Zang, *Spectral Methods in Fluid Dynamics.* Springer Series in Computational Physics, Springer Verlag, New York, 1988.

[16] J.S. Hesthaven, *A Stable Penalty Method for the Compressible Navier-Stokes Equations. II. One Dimensional Domain Decomposition Methods*, SIAM J. Sci. Comput. **18**(1997), pp.658-685.

[17] J.S. Hesthaven, *A Stable Penalty Method for the Compressible Navier-Stokes Equations. III. Multidimensional Domain Decomposition Methods*, SIAM J. Sci. Comput. 1997 - accepted.

[18] P. Fischer and D. Gottlieb, *On the Optimal Number of Subdomains for Hyperbolic Problems on Parallel Computers*, Int. J. Supercomputer Appl. High Perform. Comput. **11**(1997), pp. 65-76.

[19] I. Foster, *Designing and Building Parllel Programs: Concepts and Tools for Parallel Software Engineering.* Addison-Wesley, Reading, (1994) http://www.mcs.anl.gov/dbpp.

[20] I. Daubechies, *Orthonormal Basis of Compactly Supported Wavelets*, Comm. Pure Appl. Math. **41**(1988), pp. 909-996.

[21] G. Beylkin, *On the Representation of Operators in Bases of Compactly Supported Wavelets*, SIAM J. Numer. Anal. **29**(1992), pp.1716-1740.

[22] B. Fornberg, *Generation of Finite Difference Formulas on Arbitrarily Spaced Grids*, Math. Comp. **51**(1988), pp. 699-706.

This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.